10 2 ?0

A

PTO/SB/05 (1/98)

Please type a plus sign (+) inside this box → [+]

Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

JC544 U.S. PTO
04/14/98

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	DN37882YE
First Inventor or Application Identifier	Robert C. Meier
Title	Radio Frequency Local Area Network
Express Mail Label No.	EI075597159US

APPLICATION ELEMENTS
See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231**

1. [X] * Fee Transmittal Form *(e.g., PTO/SB/17)* *(Submit an original, and a duplicate for fee processing)*
2. [X] Specification [Total Pages 151] *(preferred arrangement set forth below)*
   - Descriptive title of the Invention
   - Cross References to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to Microfiche Appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
3. [X] Drawing(s) *(35 U.S.C. 113)* [Total Sheets 2]
4. Oath or Declaration [Total Pages 7]
   a. [ ] Newly executed (original or copy)
   b. [X] Copy from a prior application (37 C.F.R. § 1.63(d)) *(for continuation/divisional with Box 17 completed)* ***[Note Box 5 below]***
      i. [ ] DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 C.F.R. §§ 1.63(d)(2) and 1.33(b).
5. [X] Incorporation By Reference *(useable if Box 4b is checked)* The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered to be part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. [ ] Microfiche Computer Program *(Appendix)*
7. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. [ ] Computer Readable Copy
   b. [ ] Paper Copy (identical to computer copy)
   c. [ ] Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. [ ] Assignment Papers (cover sheet & document(s))
9. [ ] 37 C.F.R.§3.73(b) Statement *(when there is an assignee)* [ ] Power of Attorney
10. [ ] English Translation Document *(if applicable)*
11. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
12. [ ] Preliminary Amendment
13. [ ] Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
14. [ ] * Small Entity Statement(s) *(PTO/SB/09-12)* [ ] Statement filed in prior application, Status still proper and desired
15. [ ] Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. [ ] Other: ....

** A new statement is required to be entitled to pay small entity fees, except where one has been filed in a prior application and is being relied upon.*

17. **If a CONTINUING APPLICATION,** *check appropriate box, and supply the requisite information below and in a preliminary amendment:*

[X] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: 08 / 395,555

Prior application information: Examiner P. Huang Group / Art Unit: 2317

18. CORRESPONDENCE ADDRESS

[ ] *Customer Number or Bar Code Label* *(Insert Customer No. or Attach bar code label here)* or [X] *Correspondence address below*

Name	Gary R. Stanford				
	Akin, Gump, Strauss, Hauer & Feld, L.L.P.				
Address	816 Congress Avenue				
	Suite 1900				
City	Austin	State	Texas	Zip Code	78701
Country	US	Telephone	(512)499-6219	Fax	(512)499-6290

Name (Print/Type)	Gary R. Stanford	Registration No. (Attorney/Agent)	35,689
Signature	Gary Stanford	Date	4/14/98

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

PTO/SB/17 (12-97)
Approved for use through 9/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

FEE TRANSMITTAL

Note: Effective October 1, 1997.
Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT ($) 790.00

Complete if Known	
Application Number	Unassigned
Filing Date	4/14/98
First Named Inventor	Robert C. Meier
Group Art Unit	2317
Examiner Name	P. Huang
Attorney Docket Number	DN37882YE

METHOD OF PAYMENT (check one)

1. ☐ The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number: 01-0660

Deposit Account Name:

☒ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

☐ Charge the Issue Fee Set in 37 CFR 1.18 at the Mailing of the Notice of Allowance

2. ☒ Payment Enclosed: ☒ Check ☐ Money Order ☐ Other

FEE CALCULATION

1. FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	790	201	395	Utility filing fee	790.00
106	330	206	165	Design filing fee	
107	540	207	270	Plant filing fee	
108	790	208	395	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($) 790.00

2. CLAIMS

			Extra		Fee from below		Fee Paid
Total Claims		-20 =		X		=	
Independent Claims		- 3 =		X		=	
Multiple Dependent Claims				X		=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	22	203	11	Claims in excess of 20
102	82	202	41	Independent claims in excess of 3
104	270	204	135	Multiple dependent claim
109	82	209	41	Reissue independent claims over original patent
110	22	210	11	Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	950	217	475	Extension for reply within third month	
118	1,510	218	755	Extension for reply within fourth month	
128	2,060	228	1,030	Extension for reply within fifth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,320	241	660	Petition to revive - unintentional	
142	1,320	242	660	Utility issue fee (or reissue)	
143	450	243	225	Design issue fee	
144	670	244	335	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	790	246	395	Filing a submission after final rejection (37 CFR 1.129(a))	
149	790	249	395	For each additional invention to be examined (37 CFR 1.129(b))	

Other fee (specify) ____________

Other fee (specify) ____________

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($)

SUBMITTED BY				Complete (if applicable)	
Typed or Printed Name	Gary R. Stanford			Reg. Number	35,689
Signature	Gary Stanford	Date	4/14/98	Deposit Account User ID	01-0660

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
(Attorney Docket No. DN37882YE)

In the Application of: Meier et al. Serial No.: Unassigned Filed: Herewith For: RADIO FREQUENCY LOCAL AREA NETWORK	§ § § § § § § § § § § §	Group Art Unit: Examiner: CERTIFICATE OF EXPRESS MAILING I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to addressee" Service under 37 C.F.R. Sec. 1.10 addressed to: Box: Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231, on April 14, 1998. Express Mailing Label No.: EI075597159US Deena Beasley Deena Beasley

PRELIMINARY AMENDMENT

Box: Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

Dear Sir:

In re the above-identified patent application, which is a continuation application of pending parent application (Serial No. 08/395,555 filed February 28, 1995), filed pursuant to 37 C.F.R. 1.53, please enter the following preliminary amendment.

In the Specification:

On page 1, line 15, after the title, please cancel the present section headed "CROSS-REFERENCE TO RELATED APPLICATIONS" and substitute the following new section:

-- CROSS-REFERENCE TO RELATED APPLICATIONS

The present case is a continuation of U.S. Serial No. 08/395,555 filed February 28, 1995, which is a continuation of U.S. Serial No. 08/255,848 filed June 8, 1994 by Meier et al.

044220.0000 Austin 56353 v01

(Attorney Docket Nos. 10523US01 and DN37882YC) which is a continuation of U.S. Serial No. 07/970,411, filed November 2, 1992 by Meier et al. (Attorney Docket Nos. 92P767 and DN37882YB) which is a continuation in part of U.S. Serial No. 07/968,990, filed October 30, 1992 (Attorney Docket Nos. 92P758 and DN37882YA), which is itself a continuation in part of a pending application of Meier et al., U.S. Serial No. 07/769,425, filed October 1, 1991 (Attorney Docket Nos. 91P668 and DN37882). The pending application U.S. Serial No. 08/968,990 is also a continuation in part of pending PCT application of Mahany et al., Serial No. PCT/US92/08610, filed October 1, 1992 (Attorney Docket Nos. 92P661 and DN378882Y).

The entire disclosures of each of these applications including the drawings and appendices are incorporated herein.--

In the Claims:

Please cancel claims 1 through 14 without prejudice.

Please add the following new claims:

--15. A communication network supporting wireless communication of messages, said communication network comprising:

a first terminal node having a wireless receiver operable in a normal state;

a second terminal node having a wireless receiver operable in a power saving state;

an access point that attempts to immediately deliver messages destined for the first terminal node;

the access point attempts to deliver messages destined for the second terminal node by transmitting at predetermined intervals beacons that identify that a message awaits delivery;

the second terminal node synchronizes operation of its wireless receiver to receive the beacons from the access point; and

the second terminal node determines from the received beacons that it has a message awaiting delivery and directs further operation of its wireless receiver to receive the message.—

--16. The communication network of claim 15 wherein the first terminal node selectively operates in one of the normal mode and a power saving state and while operating in the power saving state the first terminal node synchronizes operation of its wireless receiver to receive the beacons from the access point.--

--17. The communication network of claim 15 wherein the second terminal node directs further operation of its receiver to receive the message during a time period that follows one of the received beacons.--

--18. The communication network of claim 17 wherein the time period immediately follows the one of the received beacons.--

--19. The communication network of claim 17 wherein the time period follows the one of the received beacons during an awake time window.--

--20. The communication network of claim 19 wherein the awake time window occurs an offset time following the one of the received beacons.--

--21. The communication network of claim 17 wherein the second terminal node has a wireless transmitter that is used to request the message awaiting delivery.--

--22. The communication network of claim 19 wherein the second terminal node has a wireless transmitter that is used to request that the message awaiting delivery be delivered during the awake time window.--

--23. The communication network of claim 20 wherein the second terminal node has a wireless transmitter that is used to request that the message awaiting delivery be delivered during the awake time window at the offset time, wherein the awake time window and the offset time are communicated with the request.--

--24. The communication network of claim 15 wherein the second terminal node communicates to the access point an indication of whether the second terminal node operates in the power saving state.--

--25. The communication network of claim 17 wherein the access point queues the messages awaiting delivery, and removes from the queue those of the messages awaiting delivery where delivery is unsuccessful.--

--26. The communication network of claim 25 wherein the messages awaiting delivery remain in the queue until delivery is successful or until a predetermined number of the beacons occur wherein delivery is unsuccessful.--

--27. The communication network of claim 17 wherein the second terminal node synchronizes operation of its wireless receiver to receive the beacons from the access point even when one or more of the beacons from the access point have not been received.--

--28. The communication network of claim 15 wherein the second terminal node comprises a battery-powered, roaming device.--

--29. The communication network of claim 28 wherein the access point participates in spanning tree routing to support the battery-powered, roaming device.—

--30. A communication network supporting wireless communication of messages, said communication network comprising:

a first terminal node operating in a first state;

a second terminal node operating in a second state in which attempts are made to minimize power consumption by the wireless receiver

a bridging node having a wireless transceiver to support wireless communication to the first and second terminal nodes;

the bridging node attempts to deliver messages destined for the second terminal node by transmitting at predetermined intervals beacons that identify a message awaiting delivery;

the second terminal node synchronizing operation of its wireless receiver to receive the beacons from the bridging node and determining from the received beacons that it has a message awaiting delivery and responding to an awaiting message by directing further operation of its wireless receiver to receive the message; and

the bridging node delivering messages to the first terminal node without requiring the first terminal node to determine from the beacons that it has messages awaiting delivery.--

--31. The communication network of claim 30 wherein the second terminal node directs further operation of its receiver to receive the message during a time period that follows one of the received beacons.--

--32. The communication network of claim 31 wherein the time period immediately follows the one of the received beacons.--

--33. The communication network of claim 31 wherein the time period follows the one of the received beacons during an awake time window.--

--34. The communication network of claim 34 wherein the awake time window occurs an offset time following the one of the received beacons.--

Remarks

Claims 15 - 26 are now pending in this application. These claims correspond to allowed claims in the parent case.

Respectfully submitted,

Date: April 14, 1998

By: Gary Stanford
Gary R. Stanford
Reg. No. 35,689

044220.0000 Austin 56353 v01

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
(Attorney Docket Nos. 92 P 767; DN37882YB)

TITLE: RADIO FREQUENCY LOCAL AREA NETWORK

CROSS-REFERENCE TO RELATED APPLICATION

The present application is a continuation in part of a pending application of Meier et al., U.S. Serial No. ___________, filed October 30, 1991 (Attorney Docket Nos. 92 P 758; DN37882YA), which is itself a continuation in part of a pending application of Meier et al., U.S. Serial No. 07/769,425, filed October 1, 1991 (Attorney Docket Nos. 91 P 668; DN37882). The pending application U.S. Serial No. 07/769,425 is also a continuation in part of pending PCT application of Mahany et al., Serial No. PCT/US92/08610, filed October 1, 1992 (Attorney Docket Nos. 92 P 661; DN37882Y).

The entire disclosures of each of these pending applications including the drawings and appendices are incorporated herein by reference as if set forth fully in this application.

Appendix B is a microfiche appendix containing a list of the program modules (also included as Appendix A) and the program modules themselves which comprise an exemplary computer program listing of the source code used by the network controllers and intelligent base transceivers of the present invention. The microfiche appendix has ten (10) total microfiche sheets and six hundred two (602) total frames.

BACKGROUND OF THE INVENTION

In a typical radio data communication system having one or more host computers and multiple RF terminals, communication between a host computer and

an RF terminal is provided by one or more base stations. Depending upon the application and the operating conditions, a large number of these base stations may be required to adequately serve the system. For example, a radio data communication system installed in a large factory may require dozens of base stations in order to cover the entire factory floor.

In earlier RF (Radio Frequency) data communication systems, the base stations were typically connected directly to a host computer through multi-dropped connections to an Ethernet communication line. To communicate between an RF terminal and a host computer, in such a system, the RF terminal sends data to a base station and the base station passes the data directly to the host computer. Communicating with a host computer through a base station in this manner is commonly known as hopping. These earlier RF data communication systems used a single-hop method of communication.

In order to cover a larger area with an RF data communication system and to take advantage of the deregulation of the spread-spectrum radio frequencies, later-developed RF data communication systems are organized into layers of base stations. As in earlier RF data communications systems, a typical system includes multiple base stations which communicate directly with the RF terminals and the host computer. In addition, the system also includes intermediate stations that communicate with the RF terminals, the multiple base stations, and other intermediate stations. In such a system, communication from an RF terminal to a host computer may be achieved, for example, by having the RF terminal send data to an intermediate station, the intermediate station send

the data to a base station, and the base station send the data directly to the host computer. Communicating with a host computer through more than one station is commonly known as a multiple-hop communication system.

Difficulties often arise in maintaining the integrity of such multiple-hop RF data communication systems. The system must be able to handle both wireless and hard-wired station connections, efficient dynamic routing of data information, RF terminal mobility, and interference from many different sources.

SUMMARY OF THE INVENTION

The present invention solves many of the problems inherent in a multiple-hop data communication system. The present invention comprises an RF Local-Area Network capable of efficient and dynamic handling of data by routing communications between the RF Terminals and the host computer through a network of intermediate base stations.

In one embodiment of the present invention, the RF data communication system contains one or more host computers and multiple gateways, bridges, and RF terminals. Gateways are used to pass messages to and from a host computer and the RF Network. A host port is used to provide a link between the gateway and the host computer. In addition, gateways may include bridging functions and may pass information from one RF terminal to another. Bridges are intermediate relay nodes which repeat data messages. Bridges can repeat data to and from bridges, gateways and RF terminals and are used to extend the range of the gateways.

The RF terminals are attached logically to the host computer and use a network formed by a gateway and the bridges to communicate with the host computer. To set up the network, an optimal configuration for conducting network communication spanning tree is created to control the flow of data communication. To aid understanding by providing a more visual description, this configuration is referred to hereafter as a "spanning tree" or "optimal spanning tree".

Specifically, root of the spanning tree are the gateways; the branches are the bridges; and non-bridging stations, such as RF terminals, are the leaves of the tree. Data are sent along the branches

of the newly created optimal spanning tree. Nodes in the network use a backward learning technique to route packets along the correct branches.

One object of the present invention is to route data efficiently, dynamically, and without looping. Another object of the present invention is to make the routing of the data transparent to the RF terminals. The RF terminals, transmitting data intended for the host computer, are unaffected by the means ultimately used by the RF Network to deliver their data.

It is a further object of the present invention for the network to be capable of handling RF terminal mobility and lost nodes with minimal impact on the entire RF data communication system.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a functional block diagram of an RF data communication system incorporating the RF local-area network of the present invention.

FIG. 2 is a flow diagram illustrating a bridging node's construction and maintenance of the spanning tree.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 is a functional block diagram of an RF data communication system. In one embodiment of the present invention, the RF data communication system has a host computer 10, a network controller 14 and bridges 22 and 24 attached to a data communication link 16. Also attached to the data communication link 16 is a gateway 20 which acts as the root node for the spanning tree of the RF data network of the present invention. A bridge 42 is attached to the gateway 20 through a hard-wired communication link and bridges 40 and 44 are logically attached to gateway 20 by two independent RF links. Additional bridges 46, 48, 50 and 52 are also connected to the RF Network and are shown in the FIG. 1. Note that, although shown separate from the host computer 10, the gateway 20 (the spanning tree root node) may be part of host computer 10.

The FIG. 1 further shows RF terminals 100 and 102 attached to bridge 22 via RF links and RF terminal 104 attached to bridge 24 via an RF link. Also, RF terminals 106, 108, 110, 112, 114, 116, 118, and 120 can be seen logically attached to the RF Network through their respective RF links. The RF terminals in FIG. 1 are representative of non-bridging stations. In alternate embodiments of the present invention, the RF Network could contain any type of device capable of supporting the functions needed to communicate in the RF Network such as hard-wired terminals, remote printers, stationary bar code scanners, or the like. The RF data communication system, as shown in FIG. 1, represents the configuration of the system at a discrete moment in time after the initialization of the system. The RF links, as shown, are dynamic and subject to change. For example, changes in the

structure of the RF data communication system can be caused by movement of the RF terminals and by interference that affects the RF communication links.

In the preferred embodiment, the host computer 10 is an IBM 3090, the network controller 14 is a model RC3250 of the Norand Corporation, the data communication link 16 is an Ethernet link, the nodes 20, 22, 24, 40, 42, 44, 46, 48, 50 and 52 are intelligent base transceiver units of the type RB4000 of the Norand Corporation, and the RF terminals 100, 102, 104, 106, 108, 110, 112, 114, 116, 118 and 120 are of type RT1100 of the Norand Corporation.

The optimal spanning tree, which provides the data pathways throughout the communication system, is stored and maintained by the network as a whole. Each node in the network stores and modifies information which specifies how local communication traffic should flow. Optimal spanning trees assure efficient, adaptive (dynamic) routing of information without looping.

To initialize the RF data communication system, the gateway 20 and the other nodes are organized into an optimal spanning tree rooted at the gateway 20. To form the optimal spanning tree, in the preferred embodiment the gateway 20 is assigned a status of ATTACHED and all other bridges are assigned the status UNATTACHED. The gateway 20 is considered attached to the spanning tree because it is the root node. Initially, all other bridges are unattached and lack a parent in the spanning tree. At this point, the attached gateway node 20 periodically broadcasts a specific type of polling packet referred to hereafter as "HELLO packets". The HELLO packets can be broadcast using known methods of communicating via radio frequency (RF) link or via a direct wire link.

In the preferred embodiment of the present invention, the RF link is comprised of spread-spectrum transmissions using a polling protocol. Although a polling protocol is preferred, a carrier-sense multiple-access (CSMA), busy-tone, or any other protocol might also manage the communication traffic on the RF link.

HELLO packets contain 1) the address of the sender, 2) the hopping distance that the sender is from the root, 3) a source address, 4) a count of nodes in the subtree which flow through that bridge, and 5) a list of system parameters. Each node in the network is assigned a unique network service address and a node-type identifier to distinguish between different nodes and different node types. The distance of a node from the root node is measured in hops times the bandwidth of each hop. The gateway root is considered to be zero hops away from itself.

FIG. 2 is a flow diagram illustrating a bridge's participation in the construction and maintenance of the spanning tree. At a block 201, the bridge begins the local construction of the spanning tree upon power-up. Next, at a block 203, the bridge enters the UNATTACHED state, listening for HELLO packets (also referred to as HELLO messages herein) that are broadcast.

By listening to the HELLO messages, bridges can learn which nodes are attached to the spanning tree. At a block 205, the bridge responds to a HELLO packet received by sending an ATTACH.request packet to the device that sent the received HELLO packet. The ATTACH.request packet is thereafter forwarded towards and to the root node which responds by sending an ATTACH.response packet back down towards and to the bridge.

The bridge awaits the ATTACH.response packet at a block 207. Upon receipt of the ATTACH.response packet, at a block 209, the bridge enters an ATTACHED state. Thereafter, at a block 211, the bridge begins periodically broadcasting HELLO packets and begins forwarding or relaying packets received. Specifically, between HELLO packet broadcasts, the bridge listens for HELLO, DATA, ATTACH.request and ATTACH.response packets broadcast by other devices in the communication network. Upon receiving such a packet, the bridge branches to a block 213. At the block 213, if the bridge detects that it has become detached from the spanning tree the bridge will branch back to the block 203 to establish attachment. Note that although the illustration in FIG. 2 places block 213 immediately after the block 211, the bridges functionality illustrated in block 213 is actually distributed throughout the flow diagram.

If at the block 213 detachment has not occurred, at a block 214, the bridge determines if the received packet is a HELLO packet. If so, the bridge analyzes the contents of the HELLO packet at a block 215 to determine whether to change its attachment point in the spanning tree. In a preferred embodiment, the bridge attempts to maintain attachment to the spanning tree at the node that is logically closest to the root node.

The logical distance, in a preferred embodiment, is based upon the number of hops needed to reach the root node and the bandwidth of those hops. The distance the attached node is away from the root node is found in the second field of the HELLO message that is broadcast. In another embodiment of the present invention, the bridges consider the number of nodes attached to the attached node as well as the logical

distance of the attached node from the root node. If an attached node is overloaded with other attached nodes, the unattached bridge may request attachment to the less loaded node, or to a more loaded node as described above in networks having regions of substantial RF overlap. In yet another embodiment, to avoid instability in the spanning tree, the bridge would only conclude to change attachment if the logical distance of the potential replacement is greater than a threshold value.

If no change in attachment is concluded, at a block 217 the bridge branches back to the block 211. If a determination is made to change attachment, a DETACH packet is sent to the root as illustrated at a block 219. After sending the DETACH packet, the bridge branches back to the block 205 to attach to the new spanning tree node. Note that the order of shown for detachment and attachment is only illustrative and can be reversed.

Referring back to the block 214, if the received packet (at block 211) is not a HELLO packet, the bridge branches to a block 221 to forward the received packet through the spanning tree. Afterwards, the bridge branches back to the block 211 to continue the process.

Specifically, once attached, the attached bridge begins broadcasting HELLO packets (at the block 211) seeking to have all unattached bridges (or other network devices) attach to the attached bridge. Upon receiving an ATTACH.request packet, the bridge forwards that packet toward the root node (through the blocks 211, 213, 214 and 221. On its path toward the root, each node records the necessary information of how to reach requesting bridge. This process is called "backward learning" herein, and is discussed

more fully below. As a result of the backward learning, once the root node receives the ATTACH.request packet, an ATTACH.response packet can be sent through the spanning tree to the bridge requesting attachment.

After attaching to an attached node, the newly attached bridge (the child) must determine its distance from the root node. To arrive at the distance of the child from the root node, the child adds the broadcast distance of its parent from the root node to the distance of the child from its parent. In the preferred embodiment, the distance of a child from its parent is based on the bandwidth of the data communication link. For example, if the child attaches to its parent via a hard-wired link (data rate 26,000 baud), then the distance of that communication link might equal, for example, one hop. However, if the child attaches to its parent via an RF link (data rate 9600 baud), then the distance of that communication link might correspondingly be equal 3 hops. The number of the hop corresponds directly to the communication speed of the link. This may not only take into consideration baud rate, but also such factors as channel interference.

Initially, only the root gateway node 20 is broadcasting HELLO messages and only nodes 40, 42 and 44 are within range of the HELLO messages broadcast by the gateway. Therefore, after the listening period has expired, nodes 40, 42 and 44 request attachment to the gateway node 20. The unattached nodes 40, 42, and 44 send ATTACH.request packets and the attached gateway node 20 acknowledges the ATTACH.request packets with local ATTACH.confirm packets. The newly attached bridges are assigned the status ATTACHED and begin broadcasting their own HELLO

packets, looking for other unattached bridges. Again, the remaining unattached nodes attempt to attach to the attached nodes that are logically closest to the root node. For example, node 48 is within range of HELLO messages from both nodes 40 and 42. However, node 40 is three hops, via an RF link, away from the gateway root node 20 and node 42 is only one hop, via a hard-wired link, away from the gateway root node 20. Therefore, node 48 attaches to node 42, the closest node to the gateway root node 20.

The sending of HELLO messages, ATTACH.request packets and ATTACH.confirm packets continues until the entire spanning tree is established. In addition, attached bridges may also respond to HELLO messages. If a HELLO message indicates that a much closer route to the root node is available, the attached bridge sends a DETACH packet to its old parent and an ATTACH.request packet to the closer node. To avoid instability in the system and to avoid overloading any given node, an attached bridge would only respond to a HELLO message if the hop count in a HELLO packet is greater than a certain threshold value, CHANGE_THRESHOLD. In the preferred embodiment, the value of the CHANGE_THRESHOLD equals 3. In this manner, an optimal spanning tree is formed that is capable of transmitting data without looping.

Nodes, other than the gateway root node, after acknowledging an ATTACH.request packet from a previously unattached node, will send the ATTACH.request packet up the branches of the spanning tree to the gateway root node. As the ATTACH.request packet is being sent to the gateway root node, other nodes attached on the same branch record the destination of the newly attached node in their routing entry table. When the ATTACH.request packet

reaches the gateway root node, the gateway root node returns an end-to-end ATTACH.confirm packet.

After the spanning tree is initialized, the RF terminals listen for periodically broadcasted Hello packets to determine which attached nodes are in range. After receiving HELLO messages from attached nodes, an RF terminal responding to an appropriate poll sends an ATTACH.request packet to attach to the node logically closest to the root. For example, RF terminal 110 is physically closer to node 44. However, node 44 is three hops, via an RF link, away from the gateway root node 20 and node 42 is only one hop, via a hard-wired link, away from the gateway root node 20. Therefore, RF terminal 110, after hearing HELLO messages from both nodes 42 and 44, attaches to node 42, the closest node to the gateway root node 20. Similarly, RF terminal 114 hears HELLO messages from nodes 48 and 50. Nodes 48 and 50 are both four hops away from the gateway root node 20. However, node 48 has two RF terminals 110 and 112 already attached to it while node 50 has only one RF terminal 116 attached to it. Therefore, RF terminal 114 will attach to node 50, the least busy node of equal distance to the gateway root node 20. Attaching to the least busy node proves to be the most efficient practice when the communication system has little overlap in the RF communication regions. In another embodiment, however, instead of attaching to the least busy node of equal distance to the gateway root node 20, the attachment is established with the busiest node.

The attached node acknowledges the ATTACH.request and sends the ATTACH.request packet to the gateway root node. Then, the gateway root node returns an end-to-end ATTACH.confirm packet. In this manner, the end-to-end ATTACH.request functions as a discovery

packet enabling the gateway root node, and all other nodes along the same branch, to learn the address of the RF terminal quickly. This process is called backward learning. Nodes learn the addresses of terminals by monitoring the traffic from terminals to the root. If a packet arrives from a terminal that is not contained in the routing table of the node, an entry is made in the routing table. The entry includes the terminal address and the address of the node that sent the packet. In addition, an entry timer is set for that terminal. The entry timer is used to determine when RF terminals are actively using the attached node. Nodes maintain entries only for terminals that are actively using the node for communication. If the entry timer expires due to lack of communication, the RF terminal entry is purged from the routing table.

The RF links among the RF terminals, the bridges, and the gateway are often lost. Therefore, a connection-oriented data-link service is used to maintain the logical node-to-node links. In the absence of network traffic, periodic messages are sent and received to ensure the stability of the RF link. As a result, the loss of a link is quickly detected and the RF Network can attempt to establish a new RF link before data transmission from the host computer to an RF terminal is adversely affected.

Communication between terminals and the host computer is accomplished by using the resulting RF Network. To communicate with the host computer, an RF terminal sends a data packet in response to a poll from the bridge closest to the host computer. Typically, the RF terminal is attached to the bridge closest to the host computer. However, RF terminals are constantly listening for HELLO and polling

messages from other bridges and may attach to, and then communicate with, a bridge in the table of bridges that is closer to the particular RF terminal.

Under certain operating conditions, duplicate data packets can be transmitted in the RF Network. For example, it is possible for an RF terminal to transmit a data packet to its attached node, for the node to transmit the acknowledgement frame, and for the RF terminal not to receive the acknowledgement. Under such circumstances, the RF terminal will retransmit the data. If the duplicate data packet is updated into the database of the host computer, the database would become corrupt. Therefore, the RF Network of the present invention detects duplicate data packets. To ensure data integrity, each set of data transmissions receives a sequence number. The sequence numbers are continuously incremented, and duplicate sequence numbers are not accepted.

When a bridge receives a data packet from a terminal directed to the host computer, the bridge forwards the data packet to the parent node on the branch. The parent node then forwards the data packet to its parent node. The forwarding of the data packet continues until the gateway root node receives the data packet and sends it to the host computer. Similarly, when a packet arrives at a node from the host computer directed to an RF terminal, the node checks its routing entry table and forwards the data packet to its child node which is along the branch destined for the RF terminal. It is not necessary for the nodes along the branch containing the RF terminal to know the ultimate location of the RF terminal. The forwarding of the data packet continues until the data packet reaches the final node on the branch, which

then forwards the data packet directly to the terminal itself.

Communication is also possible between RF terminals. To communicate with another RF terminal, the RF terminal sends a data packet to its attached bridge. When the bridge receives the data packet from a terminal directed to the host computer, the bridge checks to see if the destination address of the RF terminal is located within its routing table. If it is, the bridge simply sends the message to the intended RF terminal. If not, the bridge forwards the data packet to its parent node. The forwarding of the data packet up the branch continues until a common parent between the RF terminals is found. Then, the common parent (often the gateway node itself) sends the data packet to the intended RF terminal via the branches of the RF Network.

During the normal operation of the RF Network, RF terminals can become lost or unattached to their attached node. If an RF terminal becomes unattached, for whatever reason, its routing entry is purged and the RF terminal listens for HELLO or polling messages from any attached nodes in range. After receiving HELLO or polling messages from attached nodes, the RF terminal sends an ATTACH.request packet to the attached node closest to the root. That attached node acknowledges the ATTACH.request and sends the ATTACH.request packet onto the gateway root node. Then, the gateway root node returns an end-to-end ATTACH.confirm packet.

Bridges can also become lost or unattached during normal operations of the RF Network. If a bridge becomes lost or unattached, all routing entries containing the bridge are purged. The bridge then broadcasts a HELLO.request with a global bridge

destination address. Attached nodes will broadcast HELLO packets immediately if they receive an ATTACH.request packet with a global destination address. This helps the lost node re-attach. Then, the bridge enters the LISTEN state to learn which attached nodes are within range. The unattached bridge analyzes the contents of broadcast HELLO messages to determine whether to request attachment to the broadcasting node. Again, the bridge attempts to attach to the node that is logically closest to the root node. After attaching to the closest node, the bridge begins broadcasting HELLO messages to solicit ATTACH.requests from other nodes or RF terminals.

The spread-spectrum system provides a hierarchical radio frequency network of on-line terminals for data entry and message transfer in a mobile environment. The network is characterized by sporadic data traffic over multiple-hop data paths consisting of RS485 or ethernet wired links and single-channel direct sequenced spread spectrum links. The network architecture is complicated by moving, hidden, and sleeping nodes. The spread spectrum system consists of the following types of devices:

Terminal controller -- A gateway which passes messages from a host port to the RF network; and which passes messages from the network to the host port. The host port (directly or indirectly) provides a link between the controller and a "host" computer to which the terminals are logically attached.

Base station -- An intermediate relay node which is used to extend the range of the controller node. Base station-to-controller or base station-to-base station links can be wired or wireless RF.

Terminal -- Norand RF hand-held terminals, printers, etc. In addition, a controller device has a terminal component.

The devices are logically organized as nodes in an (optimal) spanning tree, with the controller at the root, internal nodes in base stations or controllers on branches of the tree, and terminal nodes as (possibly mobile) leaves on the tree. Like a sink tree, nodes closer to the root of the spanning tree are said to be "downstream" from nodes which are further away. Conversely, all nodes are "upstream" from the root. Packets are only sent along branches of the spanning tree. Nodes in the network use a "BACKWARD LEARNING" technique to route packets along the branches of the spanning tree.

Devices in the spanning tree are logically categorized as one of the following three node types:

1) Root (or root bridge) - A controller device which functions as the root bridge of the network spanning tree. In the preferred embodiment, the spanning tree has a single root node. Initially, all controllers are root candidates from which a root node is selected. This selection may be based on the hopping distance to the host, preset priority, random selection, etc.

2) Bridge - An internal node in the spanning tree which is used to "bridge" terminal nodes together into an interconnected network. The root node is also considered a bridge and the term "bridge" may be used to refer to all non-terminal nodes or all non-terminal nodes except the root, depending on the context herein. A bridge node consists of a network interface function and a routing function.

3) Terminal - leaf node in the spanning tree. A terminal node can be viewed as the software entity that terminates a branch in the spanning tree.

A controller device contains a terminal node(s) and a bridge node. The bridge node is the root node if the controller is functioning as the root bridge. A base station contains a bridge node. A terminal device contains a terminal node and must have a network interface function. A "bridging entity" refers to a bridge node or to the network interface function in a terminal.

The basic requirements of the system are the following.

a) Wired or wireless node connections.

b) Network layer transparency.

c) Dynamic/automatic network routing configuration.

d) Terminal mobility. Terminals should be able to move about the RF network without losing an end-to-end connection.

e) Ability to accommodate sleeping terminals.

f) Ability to locate terminals quickly.

g) Built-in redundancy. Lost nodes should have minimal impact on the network.

h) Physical link independence. The bridging algorithm is consistent across heterogeneous physical links.

The software for the spread-spectrum system is functionally layered as follows.

<u>Medium Access Control (MAC)</u>

The MAC layer is responsible for providing reliable transmission between any two nodes in the network (i.e. terminal-to-bridge). The MAC has a

channel access control component and a link control component. The link control component facilitates and regulates point-to-point frame transfers in the absence of collision detection. The MAC channel access control component regulates access to the network. Note that herein, the MAC layer is also referred to as the Data Link layer.

Bridging Layer

The bridging layer, which is also referred to herein as the network layer, has several functions as follows.

1. The bridging layer uses a "HELLO protocol" to organize nodes in the network into an optimal spanning tree rooted at the root bridge. The spanning tree is used to prevent loops in the topology. Interior branches of the spanning tree are relatively stable (i.e. controller and relay stations do not move often). Terminals, which are leaves on the spanning three, may become unattached, and must be reattached, frequently.

2. The bridging layer routes packets from terminals to the host, from the host to terminals, and from terminals to terminals along branches of the spanning tree.

3. The bridging layer provides a service for storing packets for SLEEPING terminals. Packets which cannot be delivered immediately can be saved by the bridging entity in a parent node for one or more HELLO times.

4. The bridging layer propagates lost node information throughout the spanning tree.

5. The bridging layer maintains the spanning tree links.

6. The bridging layer distributes network interface addresses.

Logical Link Control Layer

A logical link control layer, also known herein as the Transport layer herein, is responsible for providing reliable transmission between any two nodes in the network (i.e., terminal-to-base station). The data-link layer provides a connection-oriented reliable service and a connectionless unreliable service. The reliable service detects and discards duplicate packets and retransmits lost packets. The unreliable services provides a datagram facility for upper layer protocols which provide a reliable end-to-end data path. The data-link layer provides ISO layer 2 services for terminal-to-host application sessions which run on top of an end-to-end terminal-to-host transport protocol. However, the data-link layer provides transport (ISO layer 4) services for sessions contained within the SST network.

Higher Layers

For terminal-to-terminal sessions contained within the SST network, the data-link layer provides transport layer services and no additional network or transport layer is required. In this case, the MAC, bridging, and data-link layers discussed above can be viewed as a data-link layer, a network layer, and a transport layer, respectively. For terminal-to-host-application sessions, higher ISO layers exist on top of the SST data-link layer and must be implemented in the terminal and host computer, as required. This document does not define (or restrict) those layers. This document does discuss a fast-connect VMTP-like

transport protocol which is used for transient internal terminal-to-terminal sessions.

Specifically, a network layer has several functions, as follows.

1) The network layer uses a "hello protocol" to organize nodes in the network into an optimal spanning tree rooted at the controller. (A spanning tree is required to prevent loops in the topology.) Interior branches of the spanning tree are relatively stable (i.e., the controller and base stations do not move often). Terminals, which are leaves on the spanning tree, become unattached, and must be reattached frequently.

2) The network layer routes messages from terminals to the host, from the host to terminals, and from terminals to terminals along branches of the spanning tree.

3) The network layer provides a service for storing messages for SLEEPING terminals. Messages which cannot be delivered immediately can be saved by the network entity in a parent node for one or more hello times.

4) The network layer propagates lost node information throughout the spanning tree.

5) The network layer maintains the spanning tree links in the absence of regular data traffic.

A transport layer is responsible for establishing and maintaining a reliable end-to-end data path between transport access points in any two nodes in the network. The transport layer provides unreliable, reliable and a transaction-oriented services. The transport layer should be immune to implementation changes in the network layer.

The responsibilities of the transport layer include the following.

1) Establishing and maintaining TCP-like connections for reliable root-to-terminal data transmission.

2) Maintaining VMTP-like transaction records for reliable transient message passing between any two nodes.

3) Detecting and discarding duplicate packets.

4) Retransmitting lost packets.

Layers 1 through 4 are self-contained within the Norand RF network, and are independent of the host computer and of terminal applications. The session layer (and any higher layers) are dependent on specific applications. Therefore, the session protocol (and higher protocols) must be implemented as required. Note that a single transport access point is sufficient to handle single sessions with multiple nodes. Multiple concurrent sessions between any two nodes could be handled with a session identifier in a session header.

Network address requirements are as follows. DLC framed contain a hop destination and source address in the DLC header. network packets contain an end-to-end destination and a source address in the network header. Transport messages do not contain an address field; instead, a transport connection is defined by network layer source and destination address pairs. Multiple transport connections require multiple network address pairs.

The transport header contains a TRANSPORT ACCESS POINT identifier. DLC and network addresses are consistent and have the same format. Each node has a unique LONG ADDRESS which is programmed into the node at the factory. The long address is used only to obtain a SHORT ADDRESS from the root node.

The network entity in each node obtains a SHORT ADDRESS from the root node, which identifies the node uniquely. The network entity passes the short address to the DLC entity. Short addresses are used to minimize packet sizes.

Short addresses consist of the following. There is: an address length bit (short or long).

a spanning tree identified.

a node-type identifier. Node types are well known.

a unique multi-cast or broadcast node identifier.

The node-identifier parts of root addresses are well known and are constant. A default spanning tree identifier is well known by all nodes. A non-default spanning tree identifier can be entered into the root node (i.e., by a network administrator) and advertised to all other nodes in "hello" packets. The list of non-default spanning trees to which other nodes can attach must be entered into each node.

A node-type identifier of all 1's is used to specify all node types. A node identifier of all 1's is used to specify all nodes of the specified type. A DLC identifier of all 0's is used to specify a DLC entity which does not yet have an address. The all-0's address is used in DLC frames that are used to send and receive network ADDRESS packets. (The network entity in each node filters ADDRESS packets based on the network address.)

Short-address allocation is accomplished as follows. Short node identifiers of root nodes are well known. All other nodes must obtain a short node identifier from the root. To obtain a short address, a node send an ADDRESS request packet to the root node. The source addresses (i.e., DLC and network) in the request packet are LONG ADDRESSES. The root

maintains an address queue of used and unused SHORT ADDRESSES. If possible, the root selects an available short address, associates the short address with the long address of the requesting node, and returns the short address to the requesting node in an ADDRESS acknowledge packet. (Note that the destination address in the acknowledge packet is a long address.)

A node must obtain a (new) short address initially and whenever an ADDRESS-TIMEOUT inactivity period expires without having the node receive a packet from the network entity in the root.

The network entity in the root maintains addresses in the address queue in least recently used order. Whenever a packet is received, the source address is moved to the end of the queue. The address at the head of the queue is available for use by a requesting node if it has never been used or if it has been inactive for a MAX-ADDRESS-LIFE time period.

MAX-ADDRESS-LIFE must be larger than ADDRESS-TIMEOUT to ensure that an address is not in use by any node when it becomes available for another node. If the root receives an ADDRESS request from a source for which an entry exists in the address queue, the root simply updates the queue and returns the old address.

The network layer organizes nodes into an optimal spanning tree with the controller at the root of the tree. (Note that the spanning three identifier allows two logical trees to exist in the same coverage area.) Spanning tree organization is facilitated with a HELLO protocol which allows nodes to determine the shortest path to the root before attaching to the spanning tree. All messages are routed along branches of the spanning tree.

Nodes in the network are generally categorized as ATTACHED or UNATTACHED. Initially, only the root node

is attached. A single controller may be designated as the root, or multiple root candidates (i.e. controllers) may negotiate to determine which node is the root. Attached bridge nodes and root candidates transmit "HELLO" packets at calculated intervals. The HELLO packets include:

a) the source address, which includes the spanning tree ID).

b) a broadcast destination address.

c) a "seed" value from which the time schedule of future hello messages can be calculated.

d) a hello slot displacement time specifying an actual variation that will occur in the scheduled arrival of the very next hello message (the scheduled arrival being calculated from the "seed").

e) the distance (i.e., path cost) of the transmitter from the host. The incremental portion of the distance between a node and its parent is primarily a function of the type of physical link (i.e., ethernet, RS485, RF, or the like). If a signal-strength indicator is available, connections are biased toward the link with the best signal strength. The distance component is intended to bias path selection toward (i.e., wired) high-speed connections. Setting a minimum signal strength threshold helps prevent sporadic changes in the network. In addition, connections can be biased to balance the load (i.e., the number of children) on a parent node.

f) a pending message list. Pending message lists consist of 0 or more destination-address/message-length pairs. Pending messages for terminals are stored in the terminal's parent node.

g) a detached-node list. Detached-node lists contain the addresses of nodes which have detached

from the spanning tree. The root maintains two lists. A private list consists of all detached node addresses, and an advertised list consists of the addresses of all detached nodes which have pending transport messages. The addresses in the hello packet are equivalent to the advertised list.

An internal node learns which entries should be in its list from hello messages transmitted by its parent node. The root node builds its detached-node lists from information received in DETACH packets. Entries are included in hello messages for DETACH-MSG-LIFE hello times.

Attached notes broadcast "SHORT HELLO" messages immediately if they receive an "HELLO.request" packet with a global destination address; otherwise, attached nodes will only broadcast hello messages at calculated time intervals in "hello slots." Short hello messages do not contain a pending-message or detached-node list. Short hello messages are sent independently of regular hello messages and do not affect regular hello timing.

Unattached nodes (nodes without a parent in the spanning tree) are, initially, in an "UNATTACHED LISTEN" state. During the listen state, a node learns which attached base station/controller is closest to the root node by listening to hello messages. After the listening period expires an unattached node sends an ATTACH.request packet to the attached node closest to the root. The attached node immediately acknowledges the ATTACH.request, and send the ATTACH.request packet onto the root (controller) node. The root node returns the request as an end-to-end ATTACH.confirm packet. If the newly-attached node is a base station, the node calculates its link distance

and adds the distance to the distance of its parent before beginning to transmit hello messages.

The end-to-end ATTACH.request functions as a discovery packet, and enables the root node to learn the address of the source node quickly. The end-to-end ATTACH. request, when sent from a node to the root, does not always travel the entire distance. When a downstream node receives an ATTACH.request packet and already has a correct routing entry for the associated node, the downstream node intercepts the request and returns the ATTACH.confirm to the source node. (Note that any data piggy-backed on the ATTACH.request packet must still be forwarded to the host.) This situation occurs whenever a "new" path has more than one node in common with the "old" path.

The LISTEN state ends after MIN_HELLO hello time slots if hello messages have been received from at least one node. If no hello messages have been received the listening node waits and retries later.

An attached node may respond to a hello message from a node other than its parent (i.e., with an ATTACH.request) if the difference in the hop count specified in the hello packet exceeds a CHANGE-THRESHOLD level.

Unattached nodes may broadcast a GLOBAL ATTACH.request with a multi-cast base station destination address to solicit short hello messages from attached base stations. The net effect is that the LISTEN state may (optionally) be shortened. (Note that only attached base station or the controller may respond to ATTACH. requests.) Normally, this facility is reserved for base stations with children and terminals with transactions in progress.

ATTACH. requests contain a (possibly empty) CHILD LIST, to enable internal nodes to update their routing

tables. ATTACH.requests also contain a "count" field which indicates that a terminal may be SLEEPING. The network entity in the parent of a SLEEPING terminal con temporarily store messages for later delivery. If the count field is non-zero, the network entity in a parent node will store pending messages until 1) the message is delivered, or 2) "count" hello times have expired.

Transport layer data can be piggy-backed on an attached request packet from a terminal. (i.e., an attach request/confirm can be implemented with a bit flag in the network header of a data packet.)

Network layer routing.

All messages are routed along branches of the spanning tree. Base stations "learn" the address of terminals by monitoring traffic from terminals (i.e., to the root). When a base station receives (i.e., an ATTACH.request) packet, destined for the root, the base station creates or updates an entry in its routing table for the terminal. The entry includes the terminal address, and the address of the base station which sent the packet (i.e., the hop address). When a base station receives an upstream packet (i.e., from the root, destined for a terminal) the packet is simply forwarded to the base station which is in the routing entry for the destination. Upstream messages (i.e., to a terminal) are discarded whenever a routing entry does not exist. Downstream messages (i.e., from a terminal to the root) are simply forwarded to the next downstream node (i.e., the parent in the branch of the spanning tree.

TERMINAL-TO-TERMINAL COMMUNICATIONS is accomplished by routing all terminal-to-terminal traffic through the nearest common ancestor. In the

worst case, the root is the nearest common ancestor. A "ADDRESS SERVER" facilitates terminal-to-terminal communications (see below).

DELETING INVALID ROUTING TABLE ENTRIES is accomplished in several ways: connection oriented transport layer ensures that packets will arrive from nodes attached to the branch of the spanning tree within the timeout period, unless a node is disconnected.)

2) Whenever the DLC entity in a parent fails RETRY MAX times to send a message to a child node, the node is logically disconnected from the spanning tree, with one exception. If the child is a SLEEPING terminal, the message is retained by the network entity in the parent for "count" hello times. The parent immediately attempts to deliver the message after it sends its next hello packet. If, after "count" hello times, the message cannot be delivered, then the child is logically detached from the spanning tree. Detached node information is propagated downstream to the root node, each node in the path of the DETACH packet must adjust its routing tables appropriately according to the following rules: a) if the lost node is a child terminal node, the routing entry for the terminal is deleted and a DETACH packet is generated, b) if the node specified in DETACH packet is a terminal and the node which delivered the packet is the next hop in the path to the terminal, then the routing table entry for the terminal is deleted and the DETACH packet is forwarded, c) if the lost node is a child base station node then all routing entries which specify that base station as the next hop are deleted and a DETACH packet is generated for each lost terminal.

IN GENERAL, WHENEVER A NODE DISCOVERS THAT A TERMINAL IS DETACHED, IT PURGES ITS ROUTING ENTRY FOR THE TERMINAL. WHENEVER A NODE DISCOVERS THAT A BASE STATION IS DETACHED, IT PURGES ALL ROUTING ENTRIES CONTAINING THE BASE STATION. ONLY ENTRIES FOR UPSTREAM NODES ARE DELETED.

When DETACH packets reach the root node, they are added to a "detached list." Nodes remain in the root node's detached list until a) the node reattaches to the spanning tree, or b) the list entry times out. The detached list is included in hello messages and is propagated throughout the spanning tree.

For example, if a terminal detaches and reattaches to a different branch in the spanning tree, all downstream nodes in the new branch (quickly) "learn" the new path to the terminal. Nodes which were also in the old path change their routing tables and no longer forward packets along the old path. At least one node, the root, must be in both the old and new path. A new path is established as soon as an end-to-end attach request packet from the terminal reaches a node which was also in the old path.

4) A node (quickly) learns that it is detached whenever it receives a hello message, from any node, with its address in the associated detached list. The detached node can, optionally, send a global ATTACH.request, and then enters the UNATTACHED LISTEN state and reattaches as described above. After reattaching, the node must remain in a HOLD-DOWN state until its address is aged out of all detached lists. During the HOLD-DOWN state the node ignores detached lists.

5) A node becomes disconnected and enters the UNATTACHED LISTEN state whenever HELLO-RETRY-MAX hello messages are missed from its parent node.

6) A node enters the ATTACHED LISTEN state whenever a single hello message, from its parent, is missed. SLEEPING terminals remain awake during the ATTACHED LISTEN state. The state ends when the terminal receives a data or hello message from its parent. The terminal becomes UNATTACHED when a) its address appears in the detached list of a hello message from an ode other than its parent, or b) HELLO-RETRY-MAX hello messages are missed. The total number of hello slots spend in the LISTEN state is constant.

If a node in the ATTACHED LISTEN state discovers a path to the root which is CHANGE-THRESHOLD shorter, it can attach to the shorter path. Periodically, SLEEPING terminals must enter the ATTACHED LEARN state to discovery any changes (i.e., shorter paths) in the network topology.

Hello synchronization.

All attached non-terminal nodes broadcast periodic "hello" messages in discrete "hello slots" at calculated intervals. Base station nodes learn which hello slots are busy and refrain from transmitting during busy hello slots.

A terminal refrains from transmitting during the hello slot of its parent node and refrains from transmitting during message slots reserved in a hello message.

The hello message contains a "seed" field used in a well-known randomization algorithm to determine the next hello slot for the transmitting node and the next seed. The address of the transmitting node is used as a factor in the algorithm to guarantee randomization. Nodes can execute the algorithm i times to determine

the time (and seed) if the i-the hello message from the transmitter.

After attached, a base station chooses a random initial seed and a non-busy hello slot and broadcasts a hello message in that slot. The base station chooses succeeding hello slots by executing the randomization algorithm. If an execution of the algorithm chooses a busy slot, the next free slot is used and a hello "displacement" field indicates the offset from a calculated slot. Cumulative delays are not allowed (i.e., contention delays during the i hello transmission do not effect the time of the i+1 hello transmission).

HELLO-TIME and HELLO-SLOT-TIME values are set by the root node and flooded throughout the network in hello messages. The HELLO-SLOT-TIME value must be large enough to minimize hello contention.

A node initially synchronizes on a hello message from its parent. A SLEEPING node can power-down with an active timer interrupt to wake it just before the next expected hello message. The network entity in base station nodes can store messages for SLEEPING nodes and transmit them immediately following the hello messages. This implementation enables SLEEPING terminals to receive unsolicited messages. (Note that the network layer always tries to deliver messages immediately, before storing them.) Retries for pending messages are transmitted in a round-robin order when messages are pending for more than one destination.

Note that a child node that misses i hello messages, can calculate the time of the i+1 hello message.

Transport layer theory and implementation notes.

The transport layer provides reliable, unreliable, and transaction-oriented services. Two types of transport connections are defined: 1) a TCP-like transport connection may be explicitly requested for long-lived connections or 2) a VMTP-like connection-record may be implicitly set up for transient connections. In addition, a connectionless service is provided for nodes which support an end-to-end transport connection with the host computer.

The interfaces to the next upper (i.e., application) layer include:

CONNECT (access_point, node_name)

LISTEN (access_point)

UNITDATA (access_point, node_name, buffer, length)

SEND (handle, buffer, length)

RECEIVE (handle, buffer, length)

CLOSE (handle)

The "handle" designates the connection type, and is the connection identifier for TCP-like connections.

SEND messages require a response from the network node (root or terminal) to which the message is directed.

UNITDATA messages do not require a response. UNITDATA is used to send messages to a host which is capable of supporting end-to-end host-to-terminal transport connections.

Because the network layer provides an unreliable service, the transport layer is required to detect duplicate packets and retransmit lost packets. Detecting duplicates is facilitated by numbering transport packets with unambiguous sequence numbers.

Transport connections.

TCP-like transport connections are used for message transmission over long-lived connections. The connections may be terminal-to-root or terminal-to-terminal (i.e., base stations are not involved in the transport connection).

TCP-like transport connections are established using a 3-way handshake. Each end selects its initial sequence number and acknowledges the other end's initial sequence number during the handshake. The node which initiates the connection must wait a MAX-PACKET-LIFE time, before requesting a connection, to guarantee that initial sequence numbers are unambiguous. Sequence numbers are incremented modulo MAX-SEQ, where MAX-SEQ is large enough to insure that duplicate sequence numbers do not exist in the network. Packet types for establishing and breaking connections are defined as in TCP.

A TCP-like connection is full-duplex and a sliding window is used to allow multiple outstanding transport packets. An ARQ bit in the transport header is used to require an immediate acknowledgment from the opposite end.

VMTP-like connections are used for transient messages (i.e. terminal-to-terminal mail messages). VMTP-like connection records are built automatically. A VMTP-like connection record is built (or updated) whenever a VMTP-like transport message is received. The advantage is that an explicit connection request is not required. The disadvantage is that longer and more carefully selected sequence numbers are required. A VMTP-like connection is half-duplex. (A full-duplex connection at a higher layer can be built with two independent half-duplex VMTP-like connections.) Acknowledgments must be handled by higher layers.

Transport connections are defined by the network end-to-end destination and source addresses.

A MAX_TP_LIFE timeout is associated with transport connections. Transport connection records are purged after a MAX_TP_LIFE time expires without activity on the connection. The transport entity in a terminal can ensure that its transport connection will not be lost by transmitting an empty time-fill transport packet whenever TP_TIMEOUT time expires without activity.

The transport entity in a node stores messages for possible retransmission. Note that retransmissions may not always follow the same path (primarily) due to moving terminals and the resulting changes in the spanning tree. For example, the network entity in a parent node may disconnect a child after the DLC entity reports a message delivery failure. The child will soon discover that it is detached and will reattach to the spanning tree. Now when the transport entity (i.e. in the root) re-sends the message, it will follow the new path.

Transport message timing and sleeping terminals.

The transport entity in a terminal calculates a separate timeout for SEND and TRANSACTION operations. Initially, both timeouts are a function of the distance of the terminal from the root node.

A TCP-like algorithm is used to estimate the expected propagation delay for each message type. Messages, which require a response, are retransmitted if twice the expected propagation time expires before a response is received. SLEEPING terminals can power down for a large percentage of the expected propagation delay before waking up to receive the

response message. Note that missed messages may be stored by the network layer for "count" hello times.

Medium Access Control (MAC) theory and implementation notes.

Access to the network communications channel is regulated in several ways: executing the full CSMA algorithm (see MAC layer above). The sender retransmits unacknowledged messages until a RETRY_MAX count is exhausted.

The retry time of the DLC must be relatively short so that lost nodes can be detected quickly. When the DLC layer reports a failure to deliver a message to the network layer, the network layer can 1) save messages for SLEEPING terminals for later attempts, or 2) DETACH the node from the spanning tree. Note that most lost nodes are due to moving terminals.

The node identifier part of the DLC address is initially all 0's for all nodes except the root node. The all 0's address is used by a node to send and received data-link frames until a unique node identifier is passed to the DLC entity in the node. (The unique node identifier is obtained by the network entity.)

Address resolution.

Well-known names too are bound to network addresses in several ways:

- The network address and TRANSPORT ACCESS ID of a name server, contained in the root, is well-known by all nodes.

- A node can register a well-known name with the name server contained in the root node.

- A node can request the network access address of another application from the name server by using the well-known name of the application.

Possible extensions.

Base station-to-base station traffic could also be routed through the controller if the backward learning algorithm included base station nodes. (Each base station would simply have to remember which direction on its branch of the spanning tree to send data directed toward another base station.)

The possibility of multiple controllers is kept open by including a spanning-tree identifier in address fields. Each controller defines a unique spanning tree. A node can be in more than one spanning tree, with separate network state variables defined for each.

Thus, the preferred embodiment of the present invention describes an apparatus and a method of efficiently routing data through a network of intermediate base stations in a radio data communication system.

In alternate embodiments of the present invention, the RF Networks contain multiple gateways. By including a system identifier in the address field of the nodes, it is possible to determine which nodes are connected to which networks.

Multipath Fading.

In a preferred embodiment, the data to be sent through the RF communication link is segmented into a plurality of DATA packets and is then transmitted. Upon receipt, the DATA packets are reassembled for use or storage. Data segmentation of the RF link provides

better communication channel efficiency by reducing the amount of data loss in the network. For example, because collisions between transmissions on an RF link cannot be completely avoided, sending the data in small segments results in an overall decrease in data loss in the network, i.e., only the small segments which collide have to be re-sent.

Similarly, choosing smaller data packets for transmission also reduces the amount of data loss by reducing the inherent effects of perturbations and fluctuations found in RF communication links. In particular, RF signals are inherently subject to what is termed "multi-path fading." A signal received by a receiver is a composite of all signals that have reached that receiver by taking all available paths from the transmitter. The received signal is therefore often referred to as a "composite signal" which has a power envelope equal to the vector sum of the individual components of the multi-path signals received. If the signals making up the composite signal are of amplitudes that add "out of phase", the desired data signal decreases in amplitude. If the signal amplitudes are approximately equal, an effective null (no detectable signal at the receiver) results. This condition is termed "fading".

Normally changes in the propagation environment occur relatively slowly, i.e., over periods of time ranging from several tenths (1/10's) of seconds to several seconds. However, in a mobile RF environment, receivers (or the corresponding transmitters) often travel over some distance in the course of receiving a message. Because the signal energy at each receiver is determined by the paths that the signal components take to reach that receiver, the relative motion between the receiver and the transmitter causes the

receiver to experience rapid fluctuations in signal energy. Such rapid fluctuations can cause fading and result in the loss of data if the amplitude of the received signal falls below the sensitivity of the receiver.

Over small distances, the signal components that determine the composite signal are well correlated, i.e., there is a small probability that a significant change in the signal power envelope will occur over the distance. If a transmission of a data packet can be initiated and completed before the relative movement between the receiver and transmitter exceeds the "small distance", data loss to fading is unlikely to occur. The maximum "small distance" wherein a high degree of correlation exists is referred to hereafter as the "correlation distance".

As expressed in wavelengths of the carrier frequency, the correlation distance is one half (1/2) of the wavelength, while a more conservative value is one quarter (1/4) of the wavelength. Taking this correlation distance into consideration, the size of the data packet for segmentation purposes can be calculated. For example, at 915 MHz (a preferred RF transmission frequency), a quarter wavelength is about 8.2 centimeters. A mobile radio moving at ten (10) miles per hour, or 447 centimeters per second, travels the quarter wavelength in about 18.3 milliseconds. In such an environment, as long as the segment packet size remains under 18.3 milliseconds, fading does not pose any problems. In a preferred embodiment, five (5) millisecond data packet segments are chosen which provides a quasi-static multipath communication environment.

Duty Cycle.

In a preferred embodiment, each base station broadcasts HELLO messages about every two (2) seconds. If upon power up, two base stations choose to broadcast at the exact same broadcast, collisions between HELLO messages would occur and continue to occur in a lock-step fashion upon each broadcast. To prevent such an occurrence, each base station chooses a pseudo-random offset from the 2 second base time between HELLO messages to actually broadcast the HELLO message. For example, instead of beginning each HELLO message broadcast at exactly 2 seconds after the last, the base station might pseudo-randomly offset the 2 seconds by a negative (-) value of 0.2, yielding a broadcast at 1.8 seconds. Because every base station generates a different pseudo-random offset generation, the problem of lock-stepping collisions is avoided.

Additionally, instead of using a true randomization, a pseudo-random offset is used which bases all pseudo-random offset calculations on a seed value (the "seed"). The "seed" is broadcast in each HELLO message so that the timing of the next HELLO message may be calculated by any listening mobile terminal. The use of the seed, and pseudo random offset generation, allows the terminal to "sleep" (enter an energy and CPU saving mode) between HELLO messages and be able to "wake up" (dedicate energy and CPU concentration on RF reception) and stay awake for the minimal time needed to receive the next HELLO message. The relationship between the time that a base station must remain awake to the time it may sleep is called the "duty cycle".

Using a 2 second HELLO to HELLO message timing with a pseudo-random offset range of +/- 1/3 of a second, the preferred embodiment has achieved a very

low duty cycle. Further details of this timing can be found in the Bridge Layer Specification in Appendix E.

In addition, Appendix A provides a list of the program modules which are found in microfiche Appendix B. These modules comprise an exemplary computer program listing of the source code ("C" programming language) used by the network controllers and intelligent base transceivers of the present invention. Note that the term "AMX" found in Appendices A and B refers to the operating system software used. "AMX" is a multi-tasking operating system from KADAK Products, Ltd., Vancouver, B.C., Canada. Appendix C, D, E, F, and G provide system specifications for the SST Network Architecture, SST Network Frame Format, Bridging Layer, MAC Layer, and Physical Layer of one embodiment of the present invention.

As is evident from the description that is provided above, the implementation of the present invention can vary greatly depending upon the desired goal of the user. However, the scope of the present invention is intended to cover all variations and substitutions which are and which may become apparent from the illustrative embodiment of the present invention that is provided above, and the scope of the invention should be extended to the claimed invention and its equivalents.

WHAT IS CLAIMED IS:

1. A multi-hop data communication network having RF capability comprising:

a plurality of terminal nodes; and

a plurality of bridging nodes which dynamically create and revise communication pathways between any two nodes in the network, each of the bridging nodes independently storing and maintaining local information that specifies how communication traffic should flow through that bridging node, and the plurality of bridging nodes, together, providing a complete specification for the communication pathways in the multi-hop communication network; and

said nodes using HELLO messages with a backward learning technique independently create and maintain locally stored information to specify how communication traffic should flow through that bridging node.

2. The multi-hop data communication system of claim 1 further comprising means for offsetting the time period between HELLO message broadcasts.

3. The multi-hop data communication system of claim 2 further comprising means for calculating the time period between HELLO message broadcasts to be received.

4. The multi-hop data communication system of claim 3 wherein said means for offsetting further comprising a first pseudo-random number generator for generating an offset.

5. The multi-hop data communication system of claim 4 wherein said means for calculating further

comprising a second pseudo-random number generator used for computing the offset.

6. The multi-hop data communication system of claim 5 further comprising means for passing a seed value between said means for offsetting and said means for calculating so as to synchronize said first and second pseudo-random number generators.

7. A multi-hop data communication system having RF capability comprising:

a plurality of terminal nodes;

a plurality of bridging nodes; and

said bridging nodes further comprising, means for maintaining communication pathways between any two nodes in the network by repeatedly broadcasting messages identifying itself, means for determining the timing between the identifying message broadcasts.

8. The multi-hop data communication system of claim 7 wherein said terminal nodes further comprising means for calculating the time period between HELLO message broadcasts to be received.

9. The multi-hop data communication system of claim 8 wherein the means for determining the timing and the means for calculating the time both further comprise pseudo-random number generator using a common seed value.

10. The multi-hop data communication system of claim 9 further comprising means for passing a seed value between said means for determining the timing and the means for calculating the time.

11. In a multi-hop data communication network having a plurality of bridging nodes and RF communication capability, a plurality of terminal nodes comprising:

a RF transceiver;

means for segmenting digitally encoded data to be transferred into discrete data packets;

means responsive to said segmenting means for individually transmitting each discrete data packet; and

means for reconstructing discrete data packets into digitally encoded data.

12. The multi-hop data communication network of claim 11 wherein said terminal nodes further comprise means for digitally encoding voice signals, and means for generating audio signals from digitally encoded voice signals.

13. The multi-hop data communication network of claim 11 wherein the length of said discrete data packets are chosen based on correlation distance.

14. The multi-hop data communication network of claim 12 wherein the length of said discrete data packets are chosen based on correlation distance.

ABSTRACT

An apparatus and a method for routing data in a radio data communication system having one or more host computers, one or more intermediate base stations, and one or more RF terminals organizes the intermediate base stations into an optimal spanning-tree network to control the routing of data to and from the RF terminals and the host computer efficiently and dynamically. Communication between the host computer and the RF terminals is achieved by using the network of intermediate base stations to transmit the data.

FIG. 1

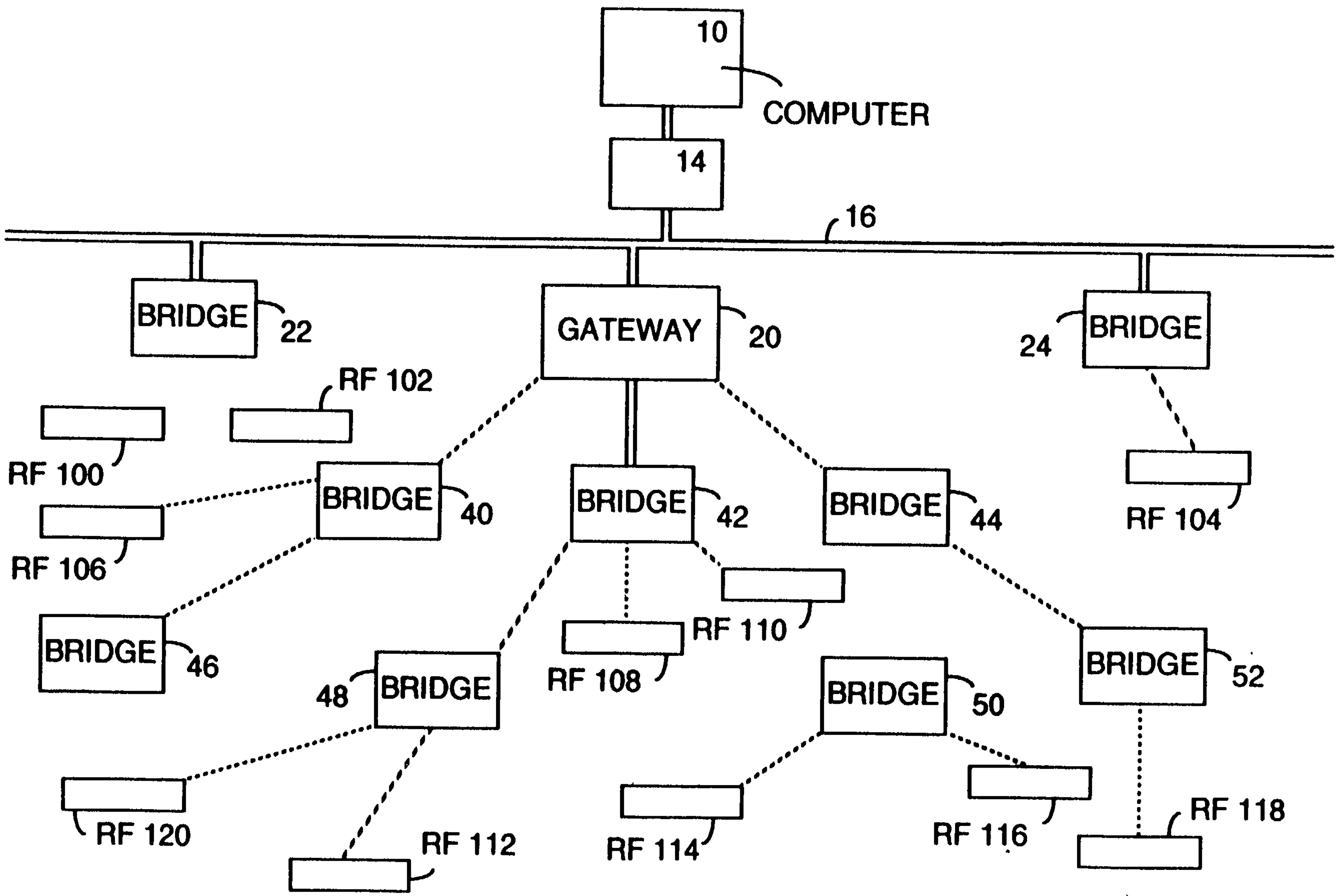
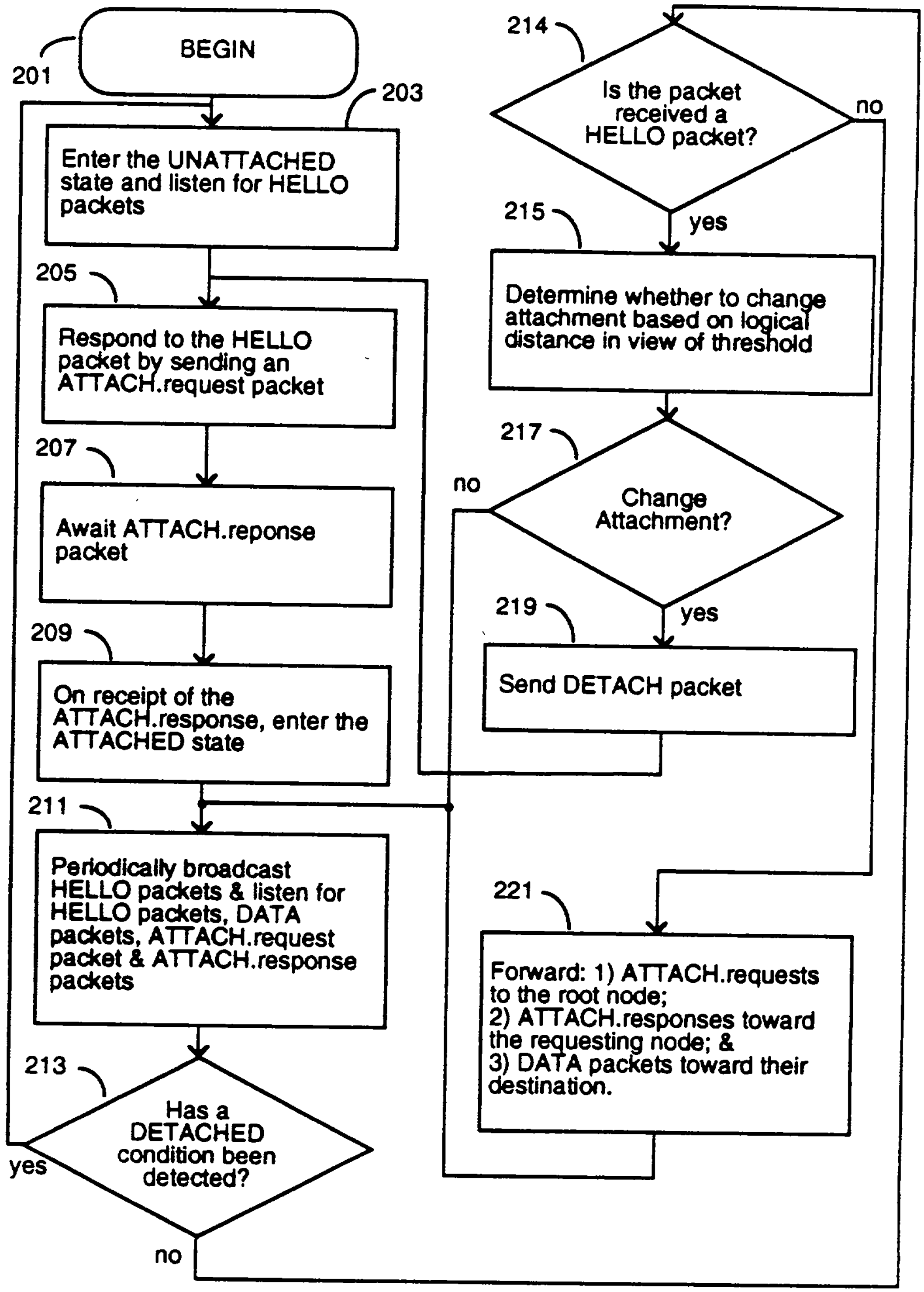

10
COMPUTER
14
16
BRIDGE
22
GATEWAY
20
BRIDGE
24
RF 102
RF 100
BRIDGE
40
BRIDGE
42
BRIDGE
44
RF 104
RF 106
RF 110
BRIDGE
46
48
BRIDGE
RF 108
BRIDGE
50
BRIDGE
52
RF 120
RF 112
RF 114
RF 116
RF 118

FIG. 2


BEGIN
201
203
Enter the UNATTACHED state and listen for HELLO packets
205
Respond to the HELLO packet by sending an ATTACH.request packet
207
Await ATTACH.reponse packet
209
On receipt of the ATTACH.response, enter the ATTACHED state
211
Periodically broadcast HELLO packets & listen for HELLO packets, DATA packets, ATTACH.request packet & ATTACH.response packets
213
Has a DETACHED condition been detected?
yes
no
214
Is the packet received a HELLO packet?
no
215
yes
Determine whether to change attachment based on logical distance in view of threshold
217
Change Attachment?
no
219
yes
Send DETACH packet
221
Forward: 1) ATTACH.requests to the root node;
2) ATTACH.responses toward the requesting node; &
3) DATA packets toward their destination.

Attorney Docket No. 92P767

CONTINUATION-IN-PART DECLARATION AND
POWER OF ATTORNEY IN PATENT APPLICATION

As a below named inventor, I hereby declare that my residence, post office address and citizenship are stated below next to my name; that I believe that I am the original and joint first inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled:

"RADIO FREQUENCY LOCAL AREA NETWORK"

described and claimed in the attached specification herewith, or as filed on November 2, 1992, as U.S. Serial No. 07/970,411.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims.

I acknowledge the duty to disclose all information known to me to be material to patentability of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119, of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)			Priority Claimed	
PCT/US92/08610	PCT	October 1, 1992	X Yes	___ No

CLAIM FOR BENEFIT OF EARLIER U.S./PCT APPLICATION(S) UNDER 35 U.S.C. 120

I hereby claim the benefit under Title 35, United States Code, Section 120, of any United States applications that are listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in those prior applications in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56, which occurred between the filing date of the prior applications and the filing date of this application:

PRIOR U.S. APPLICATIONS FOR
WHICH BENEFIT IS CLAIMED UNDER 35 U.S.C. 120

	U.S. Serial No.	U.S. Filing Date	Named Inventors
(1)	07/699,818 (Attorney Docket No. 37834)	5/13/91	R. Luse, R. Mahany, et al
(2)	07/700,704 (Attorney Docket No. 37834X)	5/14/91	R. Luse, R. Mahany, et al
(3)	07/769,425 (Attorney Docket No. 37882)	10/1/91	R. Meier, R. Luse, et al
(4)	07/790,946 (Attorney Docket No. 91P1130)	11/12/91	R. Meier
(5)	07/802,348 (Attorney Docket No. 91P1189)	12/4/91	R. Meier
(6)	07/857,603 (Attorney Docket No. 92P327)	3/30/92	R. Luse, R. Mahany, et al
(7)	07/907,927 (Attorney Docket No. 92P461)	6/30/92	R. Luse, R. Mahany, et al
(8)	07/947,102 (Attorney Docket No. 92P498)	9/14/92	R. Luse, R. Mahany, R. Meier, et al
(9)	(Attorney Docket No. 92P758)	10/30/92	R. Meier, R. Luse

I hereby appoint as my attorneys, with full powers of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

William M. Wesley Reg. No. 26,521
Herbert D. Hart, III Reg. No. 30,063
Robert W. Fieseler Reg. No. 31,826
George P. McAndrews Reg. No. 22,760
John J. Held, Jr. Reg. No. 21,061
Timothy J. Malloy Reg. No. 25,600
L. Michael Jarvis Reg. No. 27,341
Robert C. Ryan Reg. No. 29,343
Gregory J. Vogler Reg. No. 31,313
Donald J. Pochopien Reg. No. 32,167
Jean Dudek Kuelper Reg. No. 30,171
Donald P. Reynolds Reg. No. 26,220
Steven J. Hampton Reg. No. 33,707
Alejandro Menchaca Reg. No. 34,389
Priscilla F. Gallagher Reg. No. 32,223
Robert B. Polit Reg. No. 33,993
George F. Wheeler Reg. No. 28,766
D. David Hill Reg. No. 35,543
(Of McAndrews, Held & Malloy, Ltd., Northwestern Atrium Center, 500 West Madison Street, Chicago, Illinois 60661
Telephone 312/707-8889)

VanMetre Lund (Of Northbrook, Illinois) Reg. No. 16,968

R. Lewis Gable Reg. No. 22,479
(Of 2001 Jefferson Davis Highway, Arlington, VA 22202)

H. Robert Henderson (Of Des Moines, IA) Reg. No. 18,486
Michael O. Sturm (Of Des Moines, IA) Reg. No. 26,078
Sean Patrick Suiter (Of Omaha, NE) Reg. No. 34,260
Richard L. Fix (Of Des Moines, IA) Reg. No. 28,297
John E. Cepican (Of Davenport, IA) Reg. No. 26,851
William H. Wright (Of Washington, D.C.) Reg. No. 26,424
Martin G. Mullen (Of Washington, D.C.) Reg. No. 28,574
Robert J. Jondle, Ph.D., (Of Omaha, NE) Reg. No. 33,915
Daniel B. Greenwood (Of Omaha, NE) Reg. No. P35,885

Winfrid O.E. Schellin (Of Cedar Rapids, IA) Reg. No. 25,916
John H. Sherman (Of Cedar Rapids, Iowa) Reg. No. 16,909

Direct all telephone calls to:
Mr. Jack Sherman at Telephone No. 319/369-3661.

Address all correspondence to:
McAndrews, Held & Malloy, Ltd.
Northwestern Atrium Center
500 West Madison Street - 34th Floor
Chicago, IL 60661

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date 03 JANUARY 1993

Ronald L. Mahany

Full Name of joint Inventor: RONALD L. MAHANY

Residence:	Cedar Rapids	Linn	Iowa
	City	County	State

Post Office Address: 3133 Adirondack Drive NE
Cedar Rapids, Iowa 52402

Citizenship: United States of America

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any paten issued thereon.

Date 12/12/92

Robert C. Meier

Full Name of Joint Inventor: ROBERT C. MEIER

Residence:

Cedar Rapids	Linn	Iowa
City	County	State

Post Office Address: 407 Sinclair Avenue SE
Cedar Rapids, IA 52403

Citizenship: United States of America

.ENS

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any paten issued thereon.

Date 1/4/93

Ronald E. Luse

Full Name of Joint Inventor: RONALD E. LUSE

Residence:	Marion	Linn	Iowa
	City	County	State

Post Office Address: P. O. Box 545
Cedar Rapids, IA 52406

Citizenship: United States of America

.ENS